
Analysis of Dyscalculia Evidences through Artificial Intelligence Systems

F. Ferraz¹, H. Vicente^{2,3}, A. Costa² and J. Neves²

¹*Departamento de Informática, Universidade do Minho, Braga, Portugal*

²*Centro Algoritmi, Universidade do Minho, Braga, Portugal*

³*Departamento de Química, Escola de Ciências e Tecnologia, Universidade de Évora, Évora, Portugal*

E-mail: filipaferraz@gmail.com; hvicente@uevora.pt; {acosta; jneves}@di.uminho.pt

Received 5 August 2016; Accepted 17 October 2016;

Published 4 November 2016

Abstract

Dyscalculia is usually perceived of as a specific learning difficulty for mathematics or, more appropriately, arithmetic. Because definitions and diagnoses of dyscalculia are in their infancy and sometimes are contradictory. However, mathematical learning difficulties are certainly not in their infancy and are very prevalent and often devastating in their impact. Co-occurrence of learning disorders appears to be the rule rather than the exception. Co-occurrence is generally assumed to be a consequence of risk factors that are shared between disorders, for example, working memory. However, it should not be assumed that all dyslexics have problems with mathematics, although the percentage may be very high, or that all dyscalculics have problems with reading and writing. Because mathematics is very developmental, any insecurity or uncertainty in early topics will impact on later topics, hence to need to take intervention back to basics. However, it may be worked out in order to decrease its degree of severity. For example, *disMAT*, an *app* developed for *android* may help children to apply mathematical concepts, without much effort, that is turning in itself, a promising tool to dyscalculia treatment. Thus, this work will focus on the development of a *Decision Support*

Journal of Software Networking, 53–78.

doi: 10.13052/jsn2445-9739.2016.004

© 2016 River Publishers. All rights reserved.

System to estimate children evidences of dyscalculia, based on data obtained on-the-fly with *disMAT*. The computational framework is built on top of a *Logic Programming* approach to *Knowledge Representation and Reasoning*, grounded on a *Case-based* approach to computing, that allows for the handling of incomplete, unknown, or even self-contradictory information.

Keywords: Dyscalculia, Logic Programming, Knowledge Representation and Reasoning, Case-based Computing, Decision Support Systems.

1 Introduction

Dyscalculia was primarily defined as “a structural disorder of mathematical abilities” [1]. Therefore, and through some consequent studies, it can be defined as a mathematical learning disability that affects the ability to perform operations and make the proper use of arithmetic. Frequently described as “the dyslexia or blindness for numbers”, the dyscalculia is hard to be well-diagnosed, despite the incidence on 6 to 7% in the population [2]. Nevertheless, it is relevant to distinguish the type of dyscalculia under analysis:

- The person in question suffered a trauma, like an injury or a stroke, and developed this difficulty in leading with numbers, then it is called *acalculia*; and
- The disorder exists since birth, with the absence of accidents, then it is called *developmental dyscalculia*, since it will accompanied the individual through ages. This last type it is the most common and the one that is referred to as *dyscalculia* [3].

Besides affecting the realization of simple calculations with two digits and basic operations (i.e., addition, subtraction, multiplication and division), dyscalculia also influences tasks such as to distinguish left from right, to express the time, or even to count money/cash. Since this specific developmental disorder can be reflected in various areas of mathematic, dyscalculia may be set in six sub-areas, taking into account the most affected ones [1], namely:

- *Lexical dyscalculia* – troubles in reading mathematical symbols;
- *Verbal dyscalculia* – troubles in naming mathematical quantities, numbers and symbols;
- *Graphic dyscalculia* – troubles in writing mathematical symbols;
- *Operational dyscalculia* – troubles in performing mathematical operations and calculus;

- *Practognostic dyscalculia* – troubles in enumerating, manipulating and comparing real objects and pictures; and
- *Ideagnostical dyscalculia* – troubles in mental operations and in the understanding of mathematical concepts.

Regarding origins, some studies claim that the intraparietal sulcus is the area accountable for the number sense, which means that the dyscalculics present a poor activity in this brain area [4]. Conversely, geneticists propose a theory that there is a gene responsible for the transmission of this disorder through heritage, although this hypothesis is not properly proved. So, after the screening tests for dyslexia, the psychologist suggests to consult a neurologist to perform an fMRI in order to correctly diagnose if dyscalculia is the case [5].

Additionally, dyscalculia may also be classified according to the state of neurological immaturity [6, 7], namely:

- As a *former state*, related with the individuals that react favorably to therapeutically intervention;
- As a *second one*, associated to the individuals who have other learning disabilities; and
- As a *last one*, linked to the individuals that feature an intellectual deficit caused by a neurological injury(ies).

Dyscalculia is irreversible, i.e., the disorder cannot be treated, but can be worked out in order to decrease its degree of severity. This involves the methodic habit to solve exercises regarding dyscalculia's issues, like exercising memory, counting amounts, and so on, leading the subject to evolve his/her weaknesses [8].

Since dyscalculia can affect daily life, it is recommended to screen this disability in early stages of life. Afterthought, the treatment or therapeutics must commence as soon as possible, in order to provide a regular routine to dyscalculic. Therefore, the therapeutics must use attractive methods to help the children to deal with the mathematical issues [9]. In order to meet this challenge, an *android app*, *disMAT*, was developed to improve the arithmetic skills of children from the age range 5–10 through simple tasks that look at their weak spots, like measures, without much effort, obligation nor awareness, turning this support system into a promising tool [10]. *disMAT* is a simple game, where the child has to choose the level according to its difficulty and then solve every one of the nine level's tasks, trying to hit the maximum score. It presents a puzzle at the end of each level to confront the child – it is a bonus task. Besides helping the child evolve the mental calculus, this *app* is

also attractive with respect to the fact that children nowadays are keeping up with the technologic era, carrying their tablets and smartphones everywhere to entertainment.

Indeed, this paper addresses the theme of dyscalculia and describes an attempt to diagnosis the disorder using a *Case-based (CB)* approach to computing [11, 12]. The *app disMAT* was applied to a group of children within the required age range, and some parameters were recorded aiming to build up a knowledge base. To set the structure of the environment and the associate inference mechanisms, a computational framework centred on a *Logic Programming (LP)* based approach to knowledge representation and reasoning was used [13]. It may handle unknown, incomplete, or even contradictory data, information or knowledge.

2 Knowledge Representation

Many approaches to knowledge representation have been proposed using the *Logic Programming (LP)* epitome, namely in the area of *Model Theory* [14, 15], and *Proof Theory* [13, 16]. In the present work the *Proof Theoretical* approach in terms of an extension to the *LP* language is followed, where a logic program is a finite set of clauses, given in the form:

$$\begin{aligned}
 &\{ \\
 &\quad \neg p \leftarrow \text{not } p, \text{ not exception}_{p_1} \\
 &\quad p \leftarrow p_1, \dots, p_n, \text{not } q_1, \dots, \text{not } q_m \\
 &\quad ?(p_1, \dots, p_n, \text{not } q_1, \dots, \text{not } q_m) \quad (n, m \geq 0) \\
 &\quad \text{exception}_{p_1} \\
 &\quad \dots \\
 &\quad \text{exception}_{p_j} \quad (0 \leq j \leq k), \text{ being } k \text{ and integer number} \\
 &\} :: \text{scoring}_{value}
 \end{aligned}$$

where the first clause stand for predicate's closure, “,” denotes “logical and”, while “?” is a domain atom denoting falsity. The p_i , q_j , and p are classical ground literals, i.e., either positive atoms or atoms preceded by the classical negation sign \neg [13]. Indeed, \neg stands for a strong declaration that speaks for itself, and *not* denotes *negation-by-failure*, or in other words, a flop in proving a given statement, once it was not declared explicitly. Under symbols' theory, every program is associated with a set of abducibles [14, 15], given here in the

form of exceptions to the extensions of the predicates that make the program, i.e., clauses of the form:

$$exception_{p_1}, \dots, exception_{p_j} \ (0 \leq j \leq k), \text{ being } k \text{ an integer number}$$

that stand for data, information or knowledge that cannot be ruled out. On the other hand, clauses of the type:

$$? (p_1, \dots, p_n, not \ q_1, \dots, not \ q_m) \ (n, m \geq 0)$$

also named invariants or restrictions, allows one to set the context under which the universe of discourse has to be understood. The term $scoring_{value}$ stands for the relative weight of the extension of a specific *predicate* with respect to the extensions of peers ones that make the inclusive or global program.

2.1 Quantitative Knowledge

In order to set one's approach to knowledge representation, two metrics will be set, namely the Quality-of-Information (*QoI*) of a logic program that will be understood as a mathematical function that will return a truth-value ranging between 0 and 1 [17, 18], once it is fed with the extension of a given predicate, i.e., $QoI_i = 1$ when the information is *known (positive)* or *false (negative)* and $QoI_i = 0$ if the information is *unknown*. For situations where the extensions of the predicates that make the program also include *abducible* sets, its terms (or clauses) present a $QoI_i \in [0, 1]$, in the form:

$$QoI_i = 1/_{Card} \quad (1)$$

if the *abducible* set for *predicates* i and j satisfy the *invariant*:

$$? \left(\left(exception_{p_i}; exception_{p_j} \right), \neg \left(exception_{p_i}; exception_{p_j} \right) \right)$$

where “;” denotes “*logical or*” and “*Card*” stands for set cardinality, being $i \neq j$ and $i, j \geq 1$ (a pictorial view of this process is given in Figure 1(a), as a pie chart).

On the other hand, the clauses cardinality (K) will be given by $C_1^{Card} + \dots + C_{Card}^{Card}$, if there is no constraint on the possible combinations among the *abducible* clauses, being the *QoI* acknowledged as:

$$QoI_{i_{1 \leq i \leq Card}} = 1/C_1^{Card}, \dots, 1/C_{Card}^{Card} \quad (2)$$

where C_{Card}^{Card} is a card-combination subset, with *Card* elements. A pictorial view of this process is given in Figure 1(b), as a pie chart.

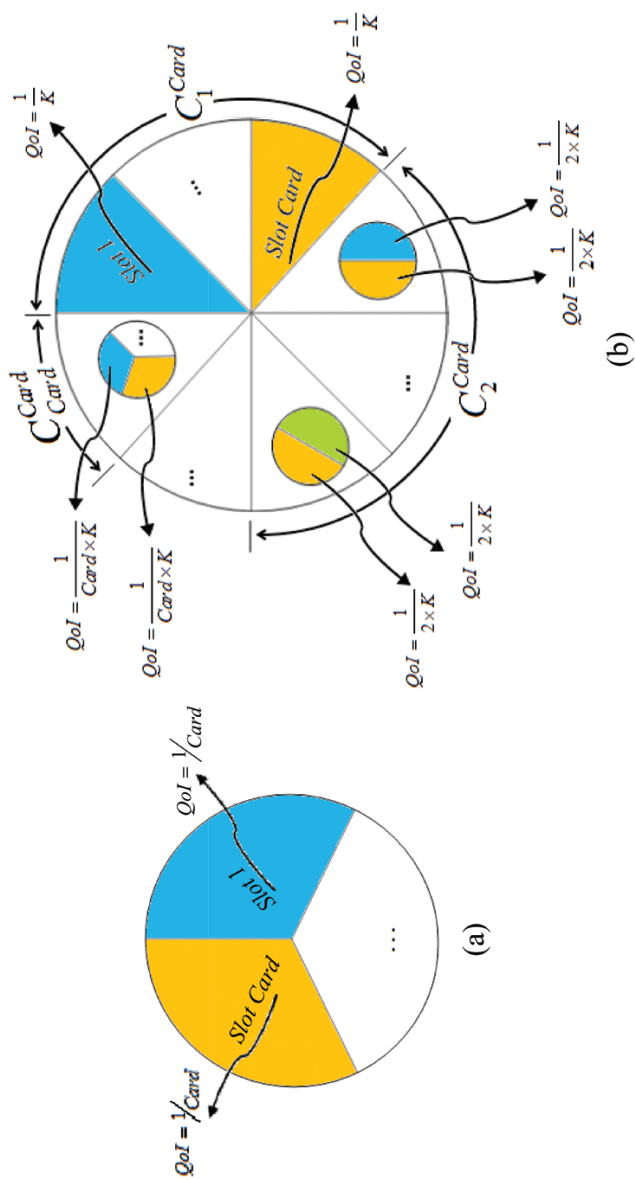


Figure 1 *QoI*s values for the abducible set for *predicate_i* with (a) and without (b) constraints on the possible combinations among the abducible clauses.

However, a term's QoI also depends on their attribute's QoI . In order to evaluate this metric, look to Figure 2, where the segment with bounds 0 and 1 stands for every attribute domain, i.e., all the attributes range in the interval $[0, 1]$. $[A, B]$ denotes the range where the unknown attributes values for a given predicate may occur (Figure 2). Therefore, the QoI of each attribute's clause is calculated using:

$$QoI_{attribute_i} = 1 - \|A - B\| \quad (3)$$

where $\|A - B\|$ stands for the modulus of the arithmetic difference between A and B . Thus, in Figure 3 is showed the QoI 's values for the abducible set for $predicate_i$.

Under this setting, another metric has to be considered, which will be denoted as DoC (*Degree-of-Confidence*), that stands for one's confidence that the argument values or attributes of the terms that make the extension of a given predicate, having into consideration their domains, are in a given interval [19]. The DoC is figured using $DoC = \sqrt{1 - \Delta l^2}$, where Δl stands for the argument interval length, which was set to the interval $[0, 1]$ (Figure 4).

Thus, the universe of discourse is engendered according to the information presented in the extensions of such predicates, according to productions of the type:

$$predicate_i - \bigcup_{1 \leq j \leq m} clause_j (((A_{x_1}, B_{x_1}) (QoI_{x_1}, DoC_{x_1})), \dots, ((A_{x_l}, B_{x_l}) (QoI_{x_l}, DoC_{x_l}))) :: QoI_j :: DoC_j \quad (4)$$

where \bigcup , m and l stand, respectively, for *set union*, the *cardinality* of the extension of $predicate_i$ and the number of attributes of each clause [19]. On the other hand, either the subscripts of the QoI_s and the DoC_s , or those of the pairs (A_s, B_s) , i.e., x_1, \dots, x_l , stand for the attributes' clauses values ranges.

2.2 Qualitative Knowledge

In present study both qualitative and quantitative data/knowledge are present. Aiming at the quantification of the qualitative part and in order to make easy the understanding of the process, it will be presented in a graphical form. Taking as an example a set of n issues regarding a particular subject (where there are k



Figure 2 Setting the QoI s of each attribute's clause.

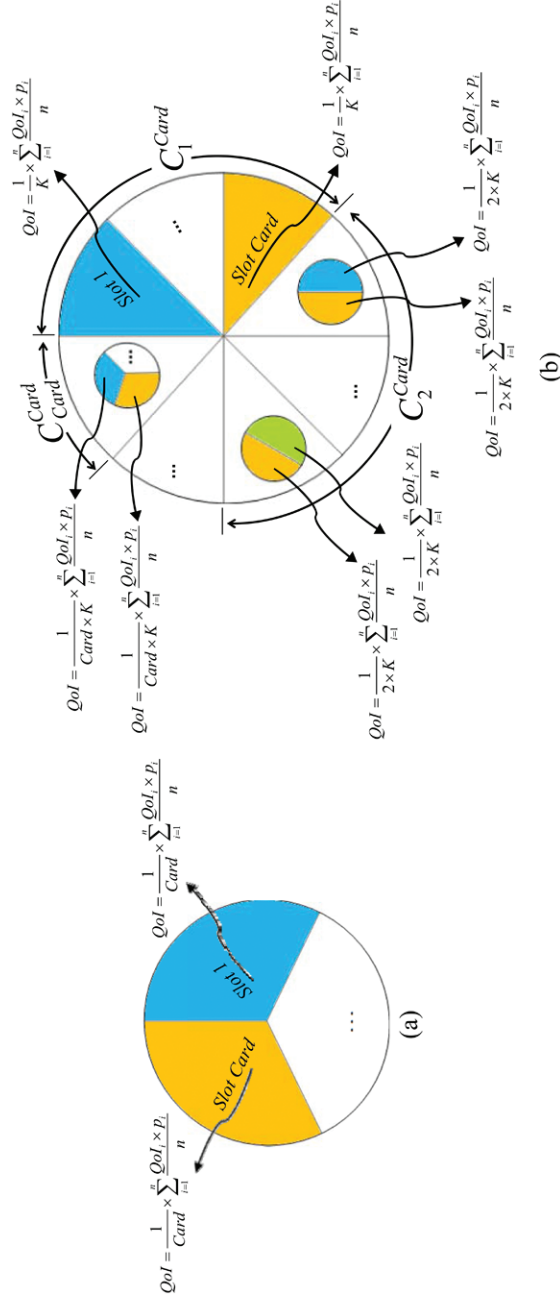


Figure 3 QoI 's values for the abducible set for $predicate_i$ with (a) and without (b) constraints on the possible combinations among the abducible clauses. $\sum_{i=1}^n (QoI_i \times p_i) / n$ denotes the QoI 's average of the attributes of each clause (or term) that sets the extension of the predicate under analysis. n and p_i stand for, respectively, for the attribute's cardinality and the relative weight of attribute p_i with respect to its peers ($\sum_{i=1}^n p_i = 1$).

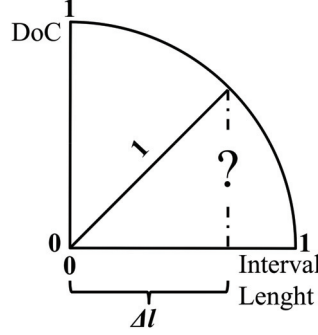


Figure 4 Evaluation of the attributes' degree of confidence.

possible choices, i.e., *none*, *low*, ..., *high* and *very high*), let us itemized an unitary area circle split into n slices (Figure 5). The marks in the axis correspond to each of the possible options. If the answer to issue 1 is *high* the correspondent area is $\pi \times \left(\sqrt{\frac{k-1}{k \times \pi}} \right)^2 / n$, i.e., $(k-1) / (k \times n)$ (Figure 5(a)). Assuming that in the issue 2 are chosen the alternatives *high* and *very high*, the correspondent area ranges between $\left[\pi \times \left(\sqrt{\frac{k-1}{k \times \pi}} \right)^2 / n, \pi \times \left(\sqrt{\frac{k}{k \times \pi}} \right)^2 / n \right]$, i.e., $[(k-1) / (k \times n), k / (k \times n)]$ (Figure 5(b)). Finally, in issue n if no alternative is ticked, all the hypotheses should be considered and the area varies in the interval $\left[0, \pi \times \left(\sqrt{\frac{k}{k \times \pi}} \right)^2 / n \right]$, i.e., $[0, k / (k \times n)]$ (Figure 5(c)). Thus, the total area is the sum of the partial ones (Figure 5(d)), i.e., $[(2k-2) / (k \times n), (3k-1) / (k \times n)]$.

3 Case-based Computing

The *Case-Based (CB)* approach to computing stands for an act of finding and justifying a solution to a given problem based on the consideration of the solutions of similar past ones, either using old solutions, or by reprocessing and generating new data or knowledge from the old ones [11, 12]. In *CB* the *cases* are stored in a *Case-base*, and those cases that are similar (or close) to a new one are used in the problem solving process.

The typical *CB* cycle (Figure 6) presents the mechanism that should be followed to have a consistent model. The first stage entails an initial description and a reprocessing of the problem's data or knowledge. The new case is defined

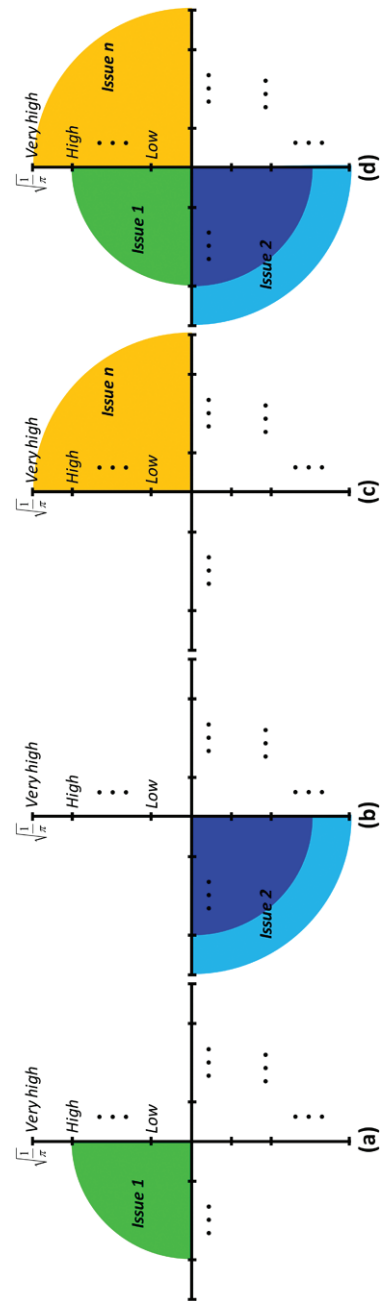


Figure 5 A view of a qualitative data/information/knowledge processing.

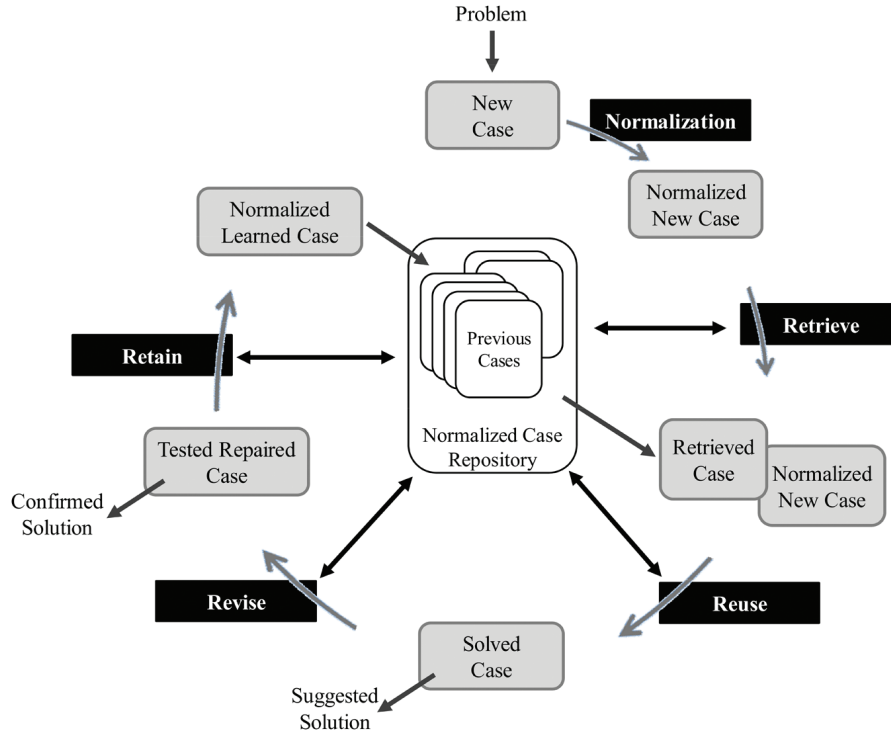


Figure 6 The typical CB cycle (adapted from [11]).

and it is used to retrieve one or more cases from the repository. At this point it is important to identify the characteristics of the new problem and retrieve cases with a higher degree of similarity to it. Thereafter, a solution to the problem emerges, on the *Reuse* phase, based on the blend of the new case with the retrieved ones. The suggested solution is reused (i.e., adapted to the new case), and a solution is provided [11, 12]. However, when adapting the solution it is crucial to have feedback from the user, since automatic adaptation in existing systems is almost impossible. This is the *Revise* stage, in which the suggested solution is tested by the user, allowing for its correction, adaptation and/or modification, originating the test repaired case that sets the solution to the new problem. The test repaired case must be correctly tested to ensure that the solution is indeed correct. Thus, one is faced with an iterative process since the solution must be tested and adapted, while the result of considering that solution is inconclusive. During the *Retain* (or *Learning*) stage the case is learned and the knowledge base is updated with the new case [11, 12].

Undeniably, despite promising results, the current *CB* systems are neither complete nor adaptable enough for all domains. In some cases, the user cannot choose the similarity(ies) method(s) used in the retrieval phase and is required to follow the system defined one(s), even if they do not meet their needs. Moreover, in real problems, the access to all necessary information is not always possible, since existent *CB* systems have limitations related to the capability of dealing, explicitly, with unknown, incomplete, and even self-contradictory information. To make a change, a different *CB* cycle was induced (Figure 7). It takes into consideration the case's *QoI* and *DoC* metrics. It also contemplates a cases optimization process present in the *Case-base*, whenever they do not comply with the terms under which a given problem as to be addressed (e.g., the expected *DoC* on a prediction was not attained). This process that uses either *Artificial Neural Networks* [20, 21], *Particle Swarm Optimization* [22] or *Genetic Algorithms* [16], just to name a few, generates a set of new cases which must be in conformity with the invariant:

$$\bigcap_{i=1}^n (B_i, E_i) \neq \emptyset \quad (5)$$

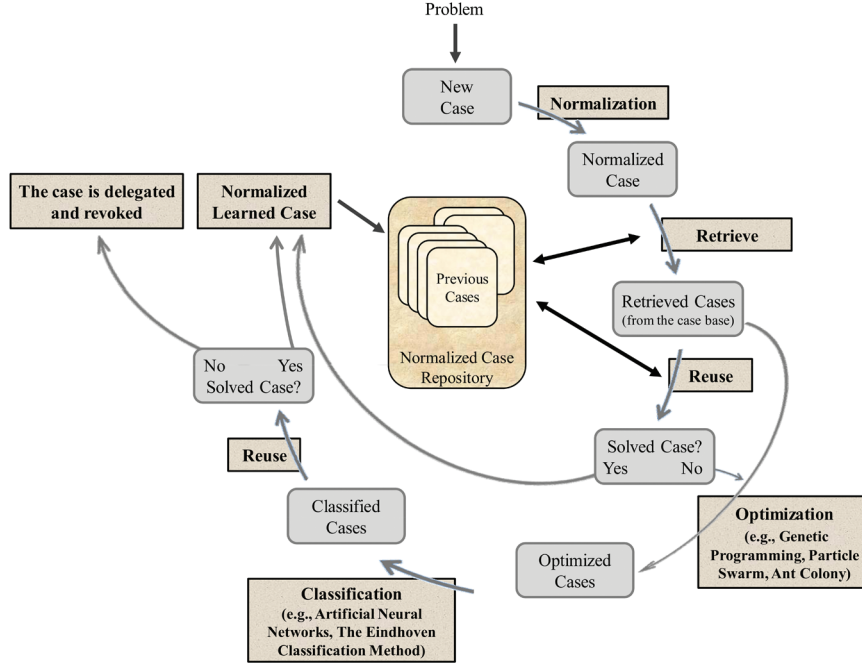


Figure 7 The *CB* cycle updated view.

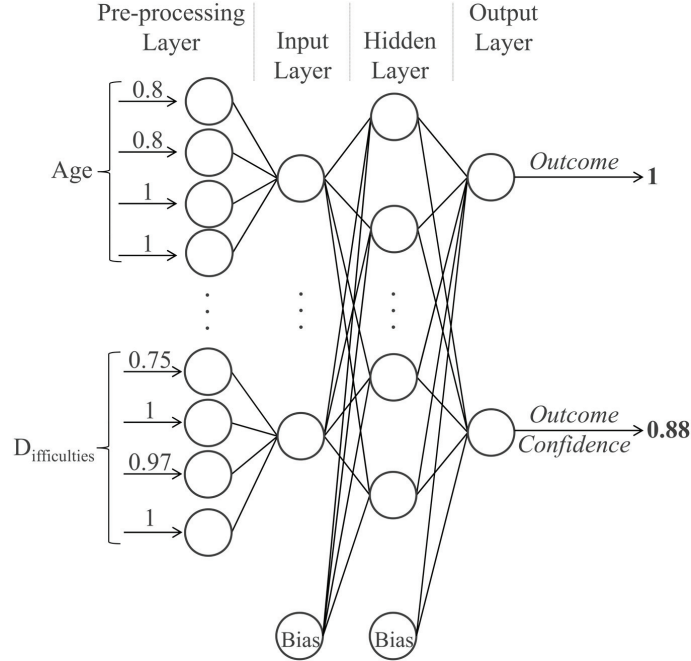


Figure 8 A case's classification procedure based on ANNs.

i.e., it states that the intersection of the attribute's values ranges for the cases' set that make the *Case-base* or their optimized counterparts (B_i) (being n its cardinality), and the ones that were object of a process of optimization (E_i), cannot be empty (Figure 7). In this *CB* cycle are used *Artificial Neural Networks* (ANNs) for case's classification in the following way:

- The extremes of the attribute's values ranges, as well as their *DoCs* and *QoIs* are fed to the ANN; and
- The outputs are given in a form that ensures that the case may be used to solve the problem (*no* (0), *yes* (1)), and a measure of the system confidence on such a result is provided (Figure 8).

4 Methods

4.1 Participants

The data was taken from the evaluation of 203 children of primary schools in the North of Portugal who played the *disMAT app* [10]. The children enrolled in

this study aged between 5 to 10 years old, with an average of 8.2 years old. The gender distribution was 42.4% and 57.6% for male and female, respectively. Forty eight students, i.e., 23.6% of the cohort, were signalized as having difficulties with numbers and mathematical concepts.

Students participated in the study voluntarily without any pressure or coercion and were informed that their grades would not be affected. Each of the participants gave an informed consent form to participate in the study. The study was conducted in compliance with the institutional guidelines. For each participant was recorded the age, the number of game levels completed, the minimum score obtained, as well as the maximum score, the response time in each of the three levels and the classification of understanding and doing difficulties through the game.

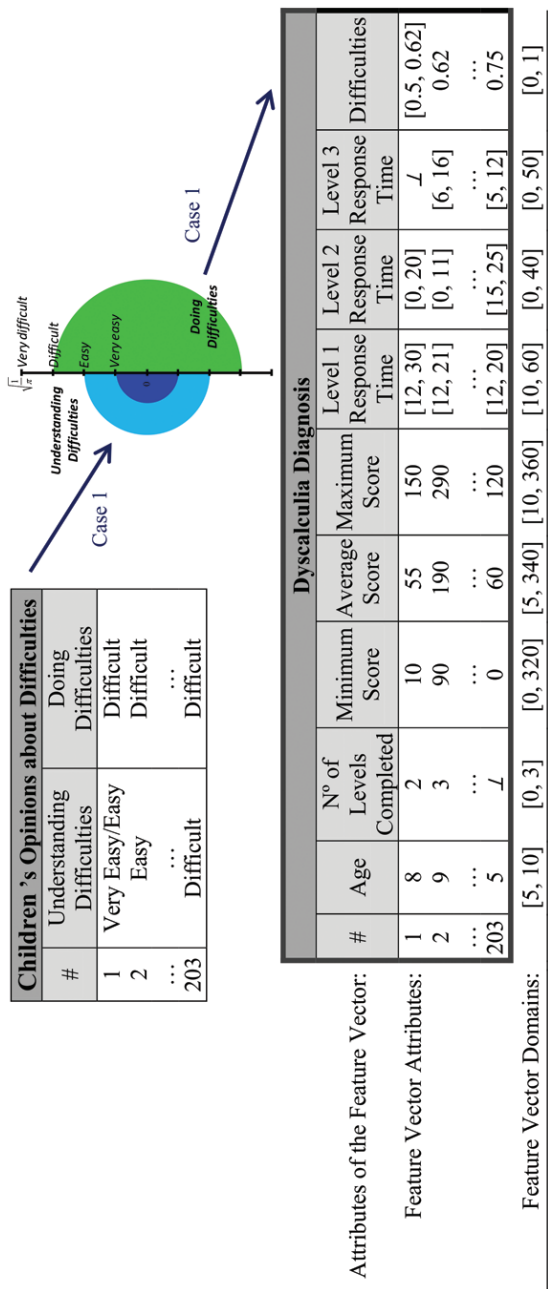
4.2 A Logic Programming Approach to Data Processing

The knowledge database is given in terms of the extensions of the tables depicted in Figure 9, which stand for a situation where one has to manage information about children evidences of dyscalculia. Under this scenario some incomplete and/or unknown data is also present. For instance, the *Level 3 Response Time* in case 1 is unknown, which is depicted by the symbol \perp , while the opinion about *Understanding Difficulties* is not conclusive (*Very Easy/Easy*).

Applying the algorithm presented in [19] to the table or relation's fields that make the knowledge base for dyscalculia diagnosis (Figure 9), and looking to the *DoCs* values obtained as described before, it is possible to set the arguments of the predicate *diagnostic* (*diag*) referred to below, whose extensions also denote the objective function with respect to the problem under analyze:

$$\begin{aligned} \text{diag: } & \text{Age, LevelsCompleted, Minimum Score, Average Score,} \\ & \text{Maximum Score, Level1 Response Time, Level2 Response Time,} \\ & \text{Level3 Response Time, Difficulties} \rightarrow \{0, 1\} \end{aligned}$$

The algorithm presented in [19] encompasses different phases. In the former one the clauses or terms that make extension of the predicate under study are established. In a second step the boundaries of the attributes intervals



are set in the interval $[0, 1]$ according to a normalization process given by the expression $(Y - Y_{min}) / (Y_{max} - Y_{min})$, where the Y_s stand for themselves. Finally, the *DoC* is evaluated as described in Section 2.1.

Exemplifying the application of the algorithm presented in [19], to a term (a *disMAT app user*) that presents the feature vector ($Age = 9, LevelsCompleted = 1, MinimumScore = 5, AverageScore = 45, MaximumScore = 60, Level1ResponseTime = [18, 30], Level2ResponseTime = \perp, Level3ResponseTime = 0, Difficulties = [0.75, 1]$), one may have:

Begin (DoCs evaluation)

The predicate's extension that sets the Universe-of-Discourse to the case (term) under observation is fixed

$$\begin{aligned}
 & \{ \\
 & \quad \neg diag \left(((A_{Age}, B_{Age}) (QoI_{Age}, DoC_{Age})), \dots, ((A_D, B_D) (QoI_D, DoC_D)) \right) \\
 & \quad \leftarrow not diag \left(((A_{Age}, B_{Age}) (QoI_{Age}, DoC_{Age})), \dots, ((A_D, B_D) (QoI_D, DoC_D)) \right) \\
 & \quad diag \left(((9, 9) (1_{[9, 9]}, DoC_{[9, 9]})), \dots, ((0.75, 1) (QoI_{[0.75, 1]}, DoC_{[0.75, 0.1]})) \right) \\
 & \quad \quad \quad :: 1 :: DoC \\
 & \quad \underbrace{[5, 10] \quad \dots \quad [0, 1]}_{attribute's domains} \\
 & \} :: 1
 \end{aligned}$$

The attribute's boundaries are set to the interval $[0, 1]$, according to a normalization process that uses the expression $(Y - Y_{min}) / (Y_{max} - Y_{min})$

$$\begin{aligned}
 & \{ \\
 & \quad \neg diag \left(((A_{Age}, B_{Age}) (QoI_{Age}, DoC_{Age})), \dots, ((A_D, B_D) (QoI_D, DoC_D)) \right) \\
 & \quad \leftarrow not diag \left(((A_{Age}, B_{Age}) (QoI_{Age}, DoC_{Age})), \dots, ((A_D, B_D) (QoI_D, DoC_D)) \right) \\
 & \quad diag \left(((0.8, 0.8) (1_{[0.8, 0.8]}, DoC_{[0.8, 0.8]})), \dots, ((0.75, 1) (1_{[0.75, 1]}, DoC_{[0.75, 1]})) \right) \\
 & \quad \quad \quad :: 1 :: DoC \\
 & \quad \underbrace{[0, 1] \quad \dots \quad [0, 1]}_{attribute's domains once normalized} \\
 & \} :: 1
 \end{aligned}$$

The DoC's values are evaluated

$$\begin{aligned}
 &\{ \\
 &\neg \text{diag } (((A_{Age}, B_{Age}) (QoI_{Age}, DoC_{Age})), \dots, ((A_D, B_D) (QoI_D, DoC_D))) \\
 &\leftarrow \text{not diag } (((A_{Age}, B_{Age}) (QoI_{Age}, DoC_{Age})), \dots, ((A_D, B_D) (QoI_D, DoC_D))) \\
 &\quad \underbrace{\text{diag } (((0.8, 0.8) (1, 1)), \dots, ((0.75, 1) (1, 0.97)))}_{\text{attribute's values ranges once normalized and} \\
 &\quad \text{respective QoI and DoC values}} :: 1 :: 0.88 \\
 &\quad \underbrace{[0, 1] \quad \dots \quad [0, 1]}_{\text{attribute's domains once normalized}} \\
 &\} :: 1 \\
 &\text{End}
 \end{aligned}$$

5 Results and Discussion

5.1 A Case-based Approach to Computing

In this section is set the model of the universe of discourse, where the computational part is based on a *CB* approach to computing. Contrasting with other problem solving tools (e.g., those that use *Decision Trees* or *Artificial Neural Networks*), relatively little work is done offline [23]. Undeniably, in almost all the situations the work is performed at query time. The main difference between this approach and the typical *CB* one relies on the fact that not only all the cases have their arguments set in the interval $[0, 1]$, a situation that is complemented with the prospect of handling incomplete, unknown, or even self-contradictory data, information or knowledge. Thus, the classic *CB* cycle was changed (Figure 7), being the *Case-base* given in terms of the pattern:

$$Case = \{ < Raw_{data}, Normalized_{data}, Description_{data} > \}$$

where the $Description_{data}$ field will not be object of attention in this study. Undeniably, when confronted with a new case, the system is able to retrieve all cases that meet such a case structure and optimize, when necessary, such a population, i.e., it considers the attributes *DoC*'s value of each case or of their optimized counterparts when analyzing similarities among them.

Thus, under the occurrence of a new case, the goal is to find similar cases in the *Case-base*. Having this in mind, the algorithm described above is applied to a new case, that presents the feature vector ($Age = 8$, $LevelsCompleted = 3$, $Minimum\ Score = \perp$, $Average\ Score = 150$, $Maximum\ Score = 330$, $Level1\ Response\ Time = 15$, $Level2\ Response\ Time = [6, 20]$, $Level3\ Response\ Time = [15, 35]$, $Difficulties = [0.88, 1]$), having in consideration that the cases retrieved from the *Case-base* satisfy the invariant presented in Equation (5), in order to ensure that the intersection of the attributes range in the cases that make the *Case-base* repository or their optimized counterparts, and the equals in the new case cannot be empty. Then, the computational process may be continued, with the outcome:

$$diag_{new} \underbrace{(((0.6, 0.6) (1, 1)), \dots, ((0.88, 1) (1, 0.99)))}_{new\ case} :: 1 :: 0.87$$

Now, the *new case* may be portrayed on the *Cartesian* plane in terms of its *QoI* and *DoC*, and by using clustering methods [24] it is feasible to identify the cluster(s) that intermingle with the *new one* (epitomized as a square in Figure 10). The *new case* is compared with every retrieved case from the clusters using a similarity function *sim*, given in terms of the average of the

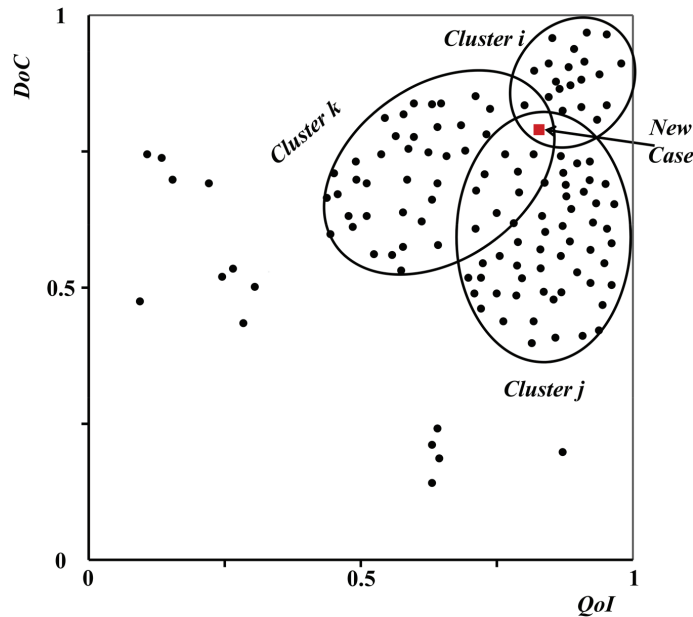


Figure 10 A case's set split into clusters.

modulus of the arithmetic difference between the arguments of each case of the selected cluster and those of the *new case*. Thus, one may have:

$$\begin{aligned} &diag_1(((0.8, 0.8)(1, 1)), \dots, ((0.75, 0.88)(1, 0.99))) :: 1 :: 0.92 \\ &diag_2(((0.6, 0.6)(1, 1)), \dots, ((0.25, 0.75)(1, 0.87))) :: 1 :: 0.84 \\ &\quad \vdots \\ &\underbrace{diag_j(((0.8, 0.8)(1, 1)), \dots, ((0.5, 0.5)(1, 1))) :: 1 :: 0.85}_{\text{normalized cases from retrieved cluster}} \end{aligned}$$

Assuming that every attribute has equal weight, for the sake of presentation, the dissimilarity between $diag_{new}$ and the $diag_1$, i.e., $diag_{new \rightarrow 1}$, may be computed as follows:

$$diag_{new \rightarrow 1}^{DoC} = \frac{\|1 - 1\| + \dots + \|0.99 - 0.99\|}{9} = 0.11$$

Thus, the similarity for $diag_{new \rightarrow 1}^{DoC}$ is set as $1 - 0.11 = 0.89$. Regarding QoI the procedure is similar, returning $diag_{new \rightarrow 1}^{QoI} = 1$. Thus, one may have:

$$diag_{new \rightarrow 1}^{QoI, DoC} = \frac{1 + 0.89}{2} = 0.94$$

These procedures should be applied to the remaining cases of the retrieved clusters in order to obtain the most similar ones, which may stand for the possible solutions to the problem.

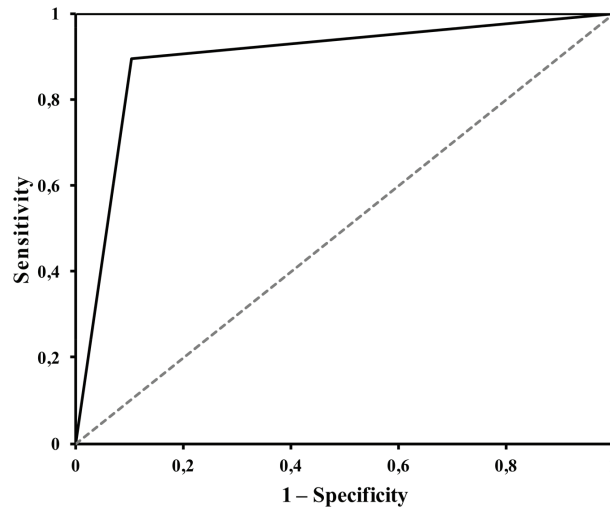
5.2 Model Validation and Performance Assessment

A common tool to evaluate the results presented by the classification models is the coincidence matrix, a matrix of size $L \times L$, where L denotes the number of possible classes. This matrix is created by matching the predicted and target values. L was set to 2 (two) in the present case. Thus, having in mind to evaluate the performance of the proposed model the dataset was divided in exclusive subsets through the ten-folds cross validation [21]. In the implementation of the respective dividing procedures, ten accomplishments were performed for each one of them. Table 1 presents the coincidence matrix of the proposed model, where the values presented denote the average of 30 (thirty) experiments. A perusal to Table 1 shows that the model accuracy was 89.7% (i.e., 182 instances correctly classified in 203). Thus, the predictions made by the presented model are satisfactory, attaining accuracy close to 90%.

Table 1 The coincidence matrix for proposed model

Target	Predictive	
	True (1)	False (0)
True (1)	43	5
False (0)	16	139

Based on coincidence matrix it is possible to compute *sensitivity*, *specificity*, *Positive Predictive Value (PPV)* and *Negative Predictive Value (NPV)* of the classifier. Briefly, *sensitivity* evaluates the proportion of true positives that are correctly identified as such, while *specificity* translates the proportion of true negatives that are correctly identified. *PPV* stands for the proportion of cases with positive results which are correctly classified while *NPV* is the proportion of cases with negative results which are successfully labeled [25, 26]. The values obtained for *sensitivity*, *specificity*, *PPV* and *NPV* were 89.6%, 89.7%, 72.9% and 96.5%, respectively. In addition, the *Receiver Operating Characteristic (ROC)* curve was considered. An *ROC* curve displays the trade-off between sensitivity and specificity. The *Area Under the Curve (AUC)* quantifies the overall ability of the test to discriminate between the output classes [25, 26]. Figure 11 depicted the *ROC* curve for the proposed model. The area under *ROC* curve close to 0.9 denoting that the model exhibits a good performance to signalize evidences of dyscalculia.

**Figure 11** The *ROC* curve for the proposed model.

6 Conclusions

This work presents an intelligent decision support system to estimate children evidences of dyscalculia based on the use of *android app disMAT*. It is centred on a formal framework based on *LP* for *Knowledge Representation and Reasoning*, complemented with a *CB* approach to computing that caters for the handling of incomplete, unknown, or even self-contradictory information. The proposed model is able to provide adequate responses once the overall accuracy is close to 90% and the area under *ROC* curve is near 0.9. The methodology followed in this work may set the basis of an overall approach to such systems, susceptible of application in different arenas. Furthermore, under this line of thinking the cases' retrieval and optimization phases were heightened when compared with existing tactics or methods. Additionally, under this scenery the users may define the weights of the cases' attributes on the fly, letting them to choose the most appropriate strategy to address the problem (i.e., it gives the user the possibility to narrow the search space for similar cases at runtime).

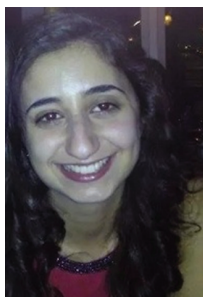
References

- [1] Kosci, L. (1974). Developmental dyscalculia. *J. Learn. Disabil.* 7, 164–177. doi: 10.1177/002221947400700309 PMID:NOPMID
- [2] Berch, D., and Mazzocco, M. (2007). *Why Is Math So Hard for Some Children? The Nature and Origins of Mathematical Learning Difficulties and Disabilities*. Baltimore, MD: Paul H. Brookes Publishing Co.
- [3] Ardila, A., and Rosselli, M. (2002). Acalculia and dyscalculia. *Neuropsychol. Rev.* 12, 179–231. doi: 10.1023/A:1020329929493 PMID:12539968
- [4] Price, G. R., and Ansari, D. (2013). Numeracy advancing education in quantitative literacy dyscalculia: characteristics, causes, and treatments. *Numeracy* 6, doi: org/10.5038/1936-4660.6.1.2
- [5] Geary, D. C. (1993). Mathematical disabilities: cognitive, neuropsychological, and genetic components. *Psychol. Bull.* 114, 345–362. doi: 10.1037/0033-2909.114.2.345 PMID:8416036
- [6] Romagnoli, G. (2008). *Dyscalculia: A Challenge in Mathematics* (in Portuguese). São Paulo: CRDA.
- [7] Barkley, R. (1982). “Guidelines for Defining Hyperactivity in Children,” in *Advances in Clinical Child Psychology*, Vol. 5, eds B. Lahey and A. Kazdin (New York: Springer), 137–180.

- [8] Michaelson, M. T. (2007). An Overview of Dyscalculia: Methods for Ascertaining and Accommodating Dyscalculic Children in the Classroom. *Aust. Math. Teach.* 63, 17–22. doi: NODOI PMID:NOPMID
- [9] Rubinsten, O., and Henik, A. (2008). Developmental dyscalculia: heterogeneity might not mean different mechanisms. *Trends Cogn. Sci.* 13, 92–99. doi: 10.5363/tits.13.6_92 PMID:19138550
- [10] Ferraz, F., and Neves, J. (2015). “A brief look into dyscalculia and supportive tools,” in *Proceedings of the 5th IEEE International Conference on E-Health and Bioengineering (EHB 2015)* (Rome: IEEE), 1–4.
- [11] Aamodt, A., and Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Commun.* 7, 39–59. doi: NODOI PMID:NOPMID
- [12] Richter, M. M., and Weber, R. O. (2013). *Case-Based Reasoning: A Textbook*. Berlin: Springer.
- [13] Neves, J. (1984). “A logic interpreter to handle time and negation in logic databases,” in *Proceedings of the 1984 annual conference of the ACM on the 5th Generation Challenge*, eds R. Muller and J. Pottmyer (New York: Association for Computing Machinery), 50–54.
- [14] Kakas, A., Kowalski, R., and Toni, F. (1998). “The role of abduction in logic programming,” in *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol. 5, eds D. Gabbay, C. Hogger, and I. Robinson (Oxford: Oxford University Press), 235–324.
- [15] Pereira, L., and Anh, H. (2009). “Evolution prospection,” in *New Advances in Intelligent Decision Technologies – Results of the First KES International Symposium IDT 2009*, Studies in Computational Intelligence, Vol. 199, ed. K. Nakamatsu (Berlin: Springer), 51–64.
- [16] Neves, J., Machado, J., Analide, C., Abelha, A., and Brito, L. (2007). “The halt condition in genetic programming,” in *Progress in Artificial Intelligence*, LNAI, Vol. 4874, eds J. Neves, M. F. Santos, and J. Machado (Berlin: Springer), 160–169.
- [17] Lucas, P. (2003). “Quality checking of medical guidelines through logical abduction,” in *Proceedings of AI-2003 (Research and Developments in Intelligent Systems XX)*, eds F. Coenen, A. Preece, and A. Mackintosh (London: Springer), 309–321.
- [18] Machado, J., Abelha, A., Novais, P., Neves, J., and Neves, J. (2008). “Quality of service in healthcare units,” in *Proceedings of the ESM 2008*, eds C. Bertelle and A. Ayesh (Ghent: Eurosis – ETI Publication), 291–298.

- [19] Fernandes, F., Vicente, H., Abelha, A., Machado, J., Novais, P., and Neves, J. (2015). “Artificial Neural Networks in Diabetes Control,” in *Proceedings of the 2015 Science and Information Conference (SAI 2015)*, (Rome: IEEE), 362–370.
- [20] Vicente, H., Couto, C., Machado, J., Abelha, A., and Neves, J. (2012). Prediction of Water Quality Parameters in a Reservoir using Artificial Neural Networks. *Int. J. Design Nat. Ecodyn.* 7, 310–319. doi: 10.1016/j.scitotenv.2014.09.005 PMID:25241206
- [21] Haykin, S. (2009). *Neural Networks and Learning Machines*. New Jersey: Pearson Education.
- [22] Mendes, R., Kennedy, J., and Neves, J. (2003). “Watch thy neighbor or how the swarm can learn from its environment,” in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS'03)*, (Rome: IEEE), 88–94.
- [23] Carneiro, D., Novais, P., Andrade, F., Zeleznikow, J., and Neves, J. (2013). Using Case-Based Reasoning and Principled Negotiation to provide decision support for dispute resolution. *Knowl. Inform. Syst.* 36, 789–826. doi: 10.1007/s10115-012-0563-0 PMID:NOPMID
- [24] Figueiredo, M., Esteves, L., Neves, J., and Vicente, H. (2016). A data mining approach to study the impact of the methodology followed in chemistry lab classes on the weight attributed by the students to the lab work on learning and motivation. *Chem. Educ. Res. Pract.* 17, 156–171. doi: 10.1039/C5RP00144G PMID:NOPMID
- [25] Florkowski, C. (2008). Sensitivity, Specificity, Receiver-Operating Characteristic (ROC) Curves and Likelihood Ratios: Communicating the Performance of Diagnostic Tests. *Clin. Biochem. Rev.* 29(Suppl. 1), S83–S87. doi: NODOI PMID:18852864
- [26] Hajian-Tilaki, K. (2013). Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. *Caspian J. Intern. Med.* 4, 627–635. doi: NODOI PMID:24009950

Biographies



F. Ferraz was born in Braga, Portugal and went to the University of Minho, where she studied Biomedical Engineering, and obtained her master degree in 2015. In 2016 she enrolled in the Doctoral Program in Biomedical Engineering, in the branch of medical informatics. Her master thesis and its theme of study in the Ph.D. are related to dyscalculia and its diagnosis and therapeutics. She is also a researcher in the Algoritmi Center of University of Minho, where she is developing software to Bosch in the line of industry 4.0. Her current interest regard Dyscalculia and Learning Disabilities, Neurology, Pediatrics, Psychiatrics, Medical Image, Internet of Things, Artificial Intelligence, Intelligent Systems, Data Mining, Knowledge Representation and Reasoning Systems, Computer Engineering, Software Engineering, Computer Science, Information Systems, and Information Technology.



H. Vicente was born in S. Martinho do Porto, Portugal and went to the University of Lisbon, where he studied Chemistry and obtained his degrees in 1988. He joined the University of Évora in 1989 and received his Ph.D. in Chemistry in 2005. He is now Auxiliary Professor at the Department of Chemistry at the University of Évora. He is a researcher at the Évora Chemistry Center and his current interests include Water Quality Control,

Lakes and Reservoirs Management, Data Mining, Knowledge Discovery from Databases, Knowledge Representation and Reasoning Systems, Evolutionary Intelligence and Intelligent Information Systems.



A. D. Costa, Ph.D., is an Assistant Professor at Department of Informatics, University of Minho, Portugal, where he develops teaching and research activities in the fields of Computer Networks and Computer Communications since 1992. As a researcher, he currently integrates the Computer Communications and Networks (CCN) research group, at Centro Algoritmi, School of Engineering, University of Minho. He graduated in Systems and Informatics Engineering in 1992, obtained a M.Sc. Degree in Informatics in 1998 and a Ph.D. Degree in Computer Science in 2006 at the same university. He participated in several research projects, supervised M.Sc. and Ph.D. students, and co-authored more than thirty peer reviewed papers in Routing Protocols, Network Services, Quality of Service, P2P, IoT and Network Management. His current research interests are in Mobile AdHoc Networks, Disruptive Delay Tolerant Networks, Named Data Networks, Indoor Positioning, Internet of Things and Future Internet.



J. Neves is Full Professor of Computer Science at Minho University, Portugal, since 1983. Jose Neves is the Deputy Director of the Division for Artificial Intelligence (AI). He received his Ph.D. in Computer Science from Heriot Watt University, Scotland, in 1983. His current research interests relate to the

areas of Knowledge Representation and Reasoning, Evolutionary Intelligence, Machine Learning, Soft Computing, aiming to construct dynamic virtual worlds of complex symbolic entities that compete against one another in which fitness is judged by one criterion alone, intelligence, here measured in terms of a process of quantification of the quality of their knowledge, leading to the development of new procedures and other metaheuristics and their application in complex tasks of optimization and model inference in distinct areas, namely in the healthcare arena (e.g., machine learning in an intensive care unit environment).